# zabbix low level discovery 自动发现

|  |  |  |  |
| --- | --- | --- | --- |
| 修订日期 | 修订人 | 修订内容 | 联系方式 |
| 2016-7-26 | 曾存凯 |  | chuck.zeng@oneoaas.com |

说明：由于做项目需要做zabbix的自动发现功能，于是昨天学习了一下zabbix自动发现的设置，学会了之后添加监控就很方便了。

本文研究测试案例：监控linux机器的端口是否存在

## 一、自动发现脚本编写

**（1）查看脚本**

shell# cat linux\_port.sh

#!/bin/bash

#function:monitor linux port status from zabbix

#mail:zck91@163.com

#version date:2016-07-26

linux.port.discory () {

port=($(cat /etc/zabbix/scripts/port.txt | grep -v "^#"))

printf '{\n'

printf '\t"data":[\n'

for ((i=0;i<${#port[@]};++i))

{

num=$(echo $((${#port[@]}-1)))

if [ "$i" != ${num} ];

then

printf "\t\t{ \n"

printf "\t\t\t\"{#SITENAME}\":\"${port[$i]}\"},\n"

else

printf "\t\t{ \n"

printf "\t\t\t\"{#SITENAME}\":\"${port[$num]}\"}]}\n"

fi

}

}

status\_code () {

port=`netstat -tunlp | grep :$1 | awk '{print $4}'| awk -F ':' '{print $2}'`

if [ -z $port ]

then

echo 1

else

echo 0

fi

}

case $1 in

linux.port.discory)

linux.port.discory

;;

status\_code)

status\_code $2

;;

\*)

echo "USage:$0 {linux.port.discory|status\_code [port]}"

;;

esac

输出的JSON格式，可以用JSON在线验证网站进行验证。

**（2）脚本验证**

shell#[root@zabbix-server-162 scripts]# bash linux\_port.sh linux.port.discory

{

"data":[

{

"{#SITENAME}":"80"},

{

"{#SITENAME}":"10050"},

{

"{#SITENAME}":"22"},

{

"{#SITENAME}":"25"},

{

"{#SITENAME}":"8080"}]}

**（3）查看port.txt**

shell# cat port.txt

80

10050

22

25

8080

## 二、自定义key配置文件

**（1）查看conf的key文件**

shell# cat userparameter\_linux\_port.conf

UserParameter=linux.port.discovery,/etc/zabbix/scripts/linux\_port.sh linux.port.discory

UserParameter=linux.port.status[\*],/etc/zabbix/scripts/linux\_port.sh status\_code $1

**（2）zabbix-server测试**

测试语句如下：

shell# zabbix\_get -s 10.0.1.162 -k linux.port.status[10050]

0 ====>> 上面脚本定义了如果是正常的即为0，如果不正常则为1，这样方便后面做trigger。

shell# zabbix\_get -s 10.0.1.162 -k linux.port.status[8080]

1 ====>> 这里检测到状态为1，则显示不正常。

至此，脚本、客户端配置文件完成。

对以上修改的文件列个清单，具体如下。

/etc/zabbix/zabbix\_agentd.conf #Agent 配置文件

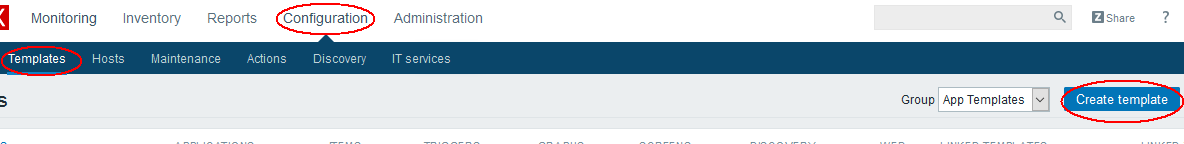
/etc/zabbix/scripts/linux\_port.sh #权限设置755

/etc/zabbix/scripts/port.txt #端口存放路径文件

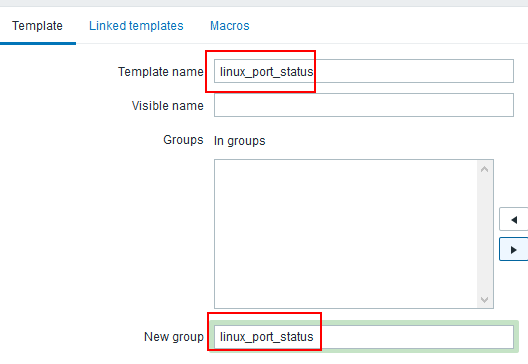
/etc/zabbix/zabbix\_agentd.conf.d/userparameter\_nginx\_port.conf #子配置文件

## 三、WEB页面配置Low level discovery

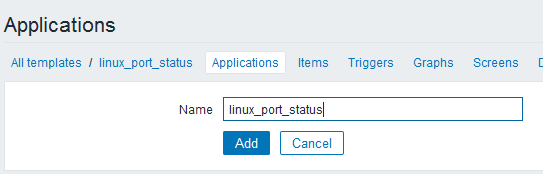
**（1）登陆zabbix页面，创建模板**



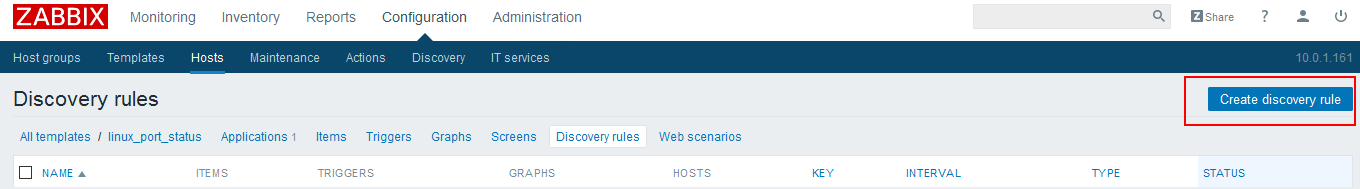
**（2）创建模板名称**

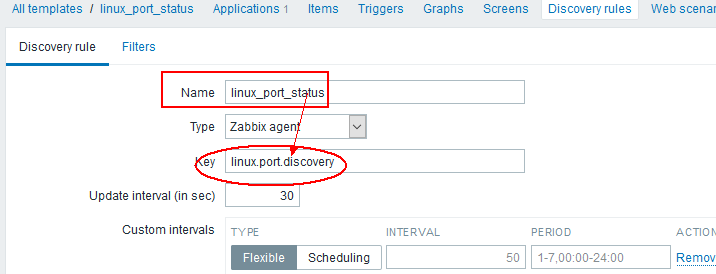


**（3）创建Application组名称**

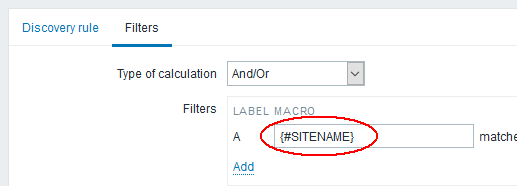


**（4）创建自动发现规则Discovery rules**

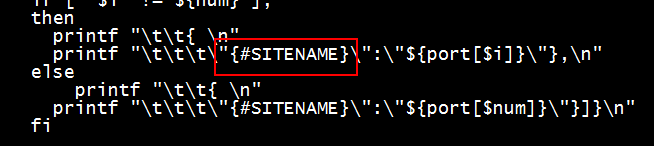




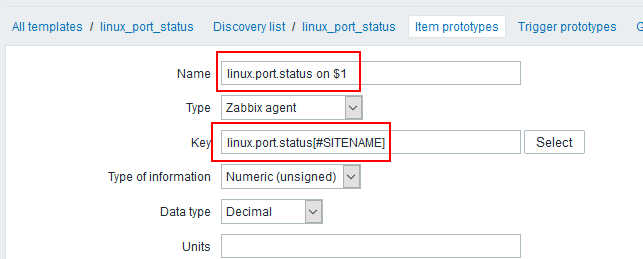
注意key一定要写自己刚刚在脚本定义的linux.port.discovery,要不然获取不到值的。



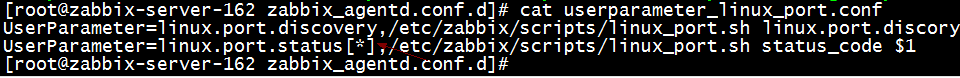
这里我们需要设置一个变量，后面无需填写，因为我们在脚本里面定义的端口变量为SITENAME



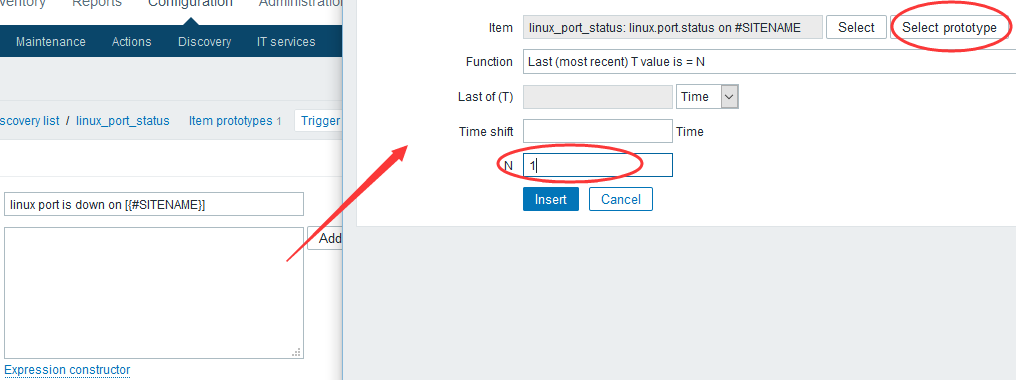
**（5）创建自动发现的item**



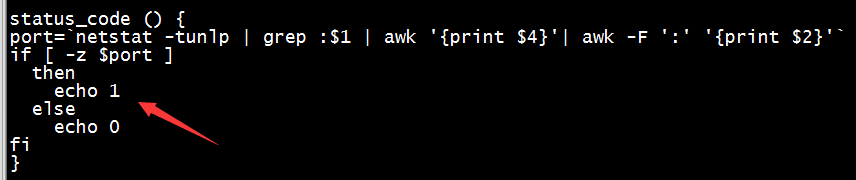
这里对应的刚刚的配置文件定义的key

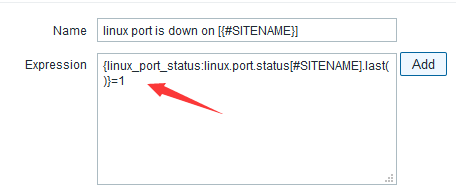


**（6）设置trigger prototypes**



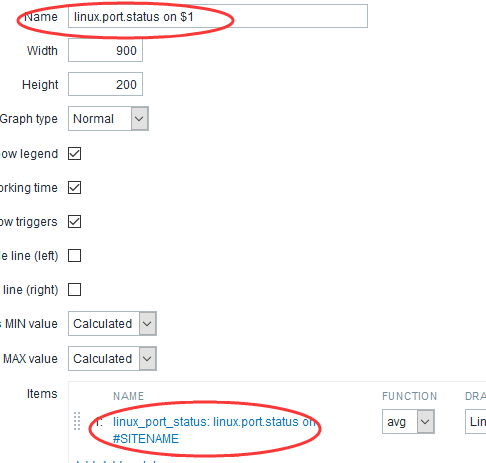
这个值我们需要设置成1，如果是1则告警。





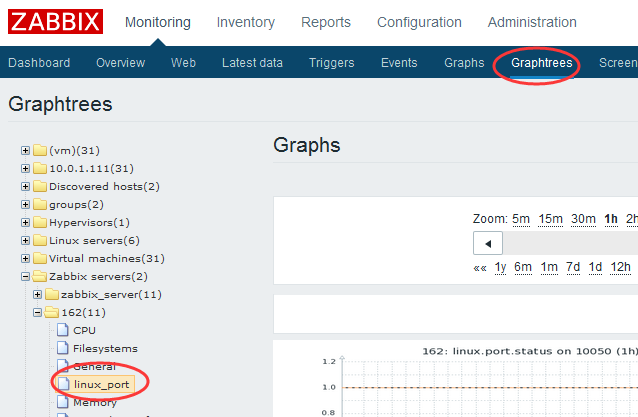
trigger添加完成

**（7）设置graph**

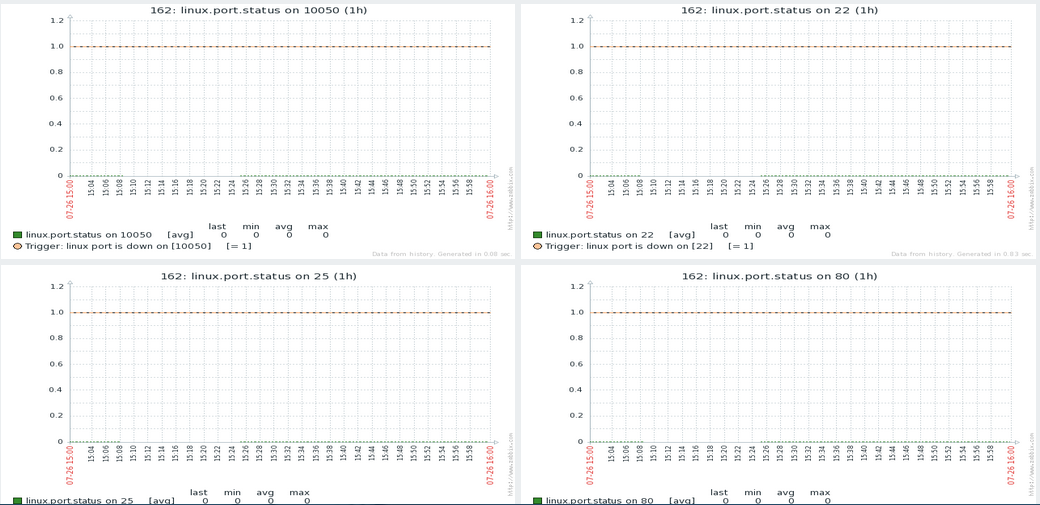


## 四、检验自动发现是否添加成功

**（1）检验是否有这个Application**



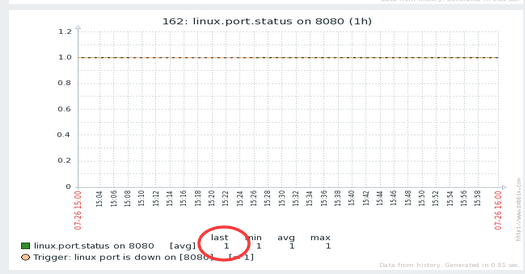
**(2) 检验是否有值**

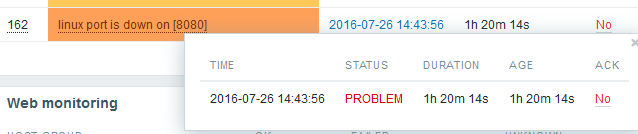


可以发现都是有值的，是0就表示正常。

**（3）验证triger是否告警**

刚刚我们设置了一个8080端口是没有的，那状态应该为1，会出现图像并且告警，我们来验证是否成功。





通过查看图像，和告警我们看到自动添加已经成功。

总结：到此我们的自动发现已经都添加完毕，有些地方需要自己去好好琢磨，例如脚本需要好好研究一下，才能够明白其中的意思。里面涉及到一个graphtree，这个是由oneoaas团队开发的产品，此功能已经开源，有需要的朋友可以联系我们oneoaas团队咨询其他关于zabbix监控项目。